CPSC 589
Seminar in Computer Science
Agent-based Computing

Chang-Hyun Jo
Department of Computer Science
California State University Fullerton
jo@ecs.fullerton.edu

Contents
- Agent-based Computing
- Agent-based Software Engineering
- Agent-based Programming Languages
- Agent-based Applications

Agents
- An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future [Franklin and Graesser 1996].

Wooldrige and Jennings [1995]
- An agent is a hardware or (more usually) software-based computer system that enjoys the following properties:
  - Autonomy
  - Social ability
  - Reactivity (Perceptive)
  - Pro-activeness:

Agents
- Autonomy
  - Agent operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state.
- Social ability
  - Agents interact with other agents (and possibly humans) via some kind of agent-communication language.

Agents
- Reactivity (Perceptive)
  - Agents perceive their environment, (which may be the physical world, a user via a GUI, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it.
- Pro-activeness:
  - Agent do not simply act in response to their environment, they are able to exhibit goal-directed behavior by taking the initiative.
Properties of Agents [Franklin and Graesser 1996]

<table>
<thead>
<tr>
<th>Property</th>
<th>Other Names</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactive</td>
<td>Setting &amp; acting</td>
<td>Responds in a timely fashion to changes in the environment</td>
</tr>
<tr>
<td>Autonomous</td>
<td></td>
<td>Can maintain control over its own actions</td>
</tr>
<tr>
<td>Goal-oriented</td>
<td>Pre-active</td>
<td>Does not simply act in response to the environment</td>
</tr>
<tr>
<td>Temporarily continuous</td>
<td></td>
<td>Is a continuously running process</td>
</tr>
<tr>
<td>Communicative</td>
<td>Socially able</td>
<td>Communicate with other agents, perhaps including people</td>
</tr>
<tr>
<td>Learning</td>
<td>Adaptive</td>
<td>Changes its behavior based on its previous experience</td>
</tr>
<tr>
<td>Mobile</td>
<td></td>
<td>Able to transport itself from one machine to another</td>
</tr>
<tr>
<td>Flexible</td>
<td></td>
<td>Actions are not scripted</td>
</tr>
<tr>
<td>Character</td>
<td></td>
<td>Is believable, personality, and emotional state</td>
</tr>
</tbody>
</table>

Classification of Agents

- KidSim Agent [Smith, Cypher, and Sponhrer 1994]: Is dedicated to a specific purpose – task-specific agent
- Hayes-Roth Agent [Hayes-Roth 1995]: Reasons to interpret perceptions, solve problems, draw inferences, and determine actions – reasoning agent
- IBM Agent [IBM ??]: Carries out some set of operations on behalf of a user or another program – task-specific agent

Classification of Agents

- Woorldridge & Jennings Agent [Woorldridge & Jennings 1995]: Interacts with other agents (and possibly humans) via some kind of agent-communication language – communicative agent
- SodaBot Agent [Cohen 1994]: Engages dialog, and negotiates and coordinates transfer of information – negotiating, information agent

Classification of Agents

- Julia Agent [Mauldin 1993]: Used in MUD (Multiple User Dimension, Multiple User Dungeon, or Multiple User Dialogue), a computer program which users can log into and explore - entertaining agent

Classification of Agents

Multiagent

- Agents must work cooperatively with other agents in a heterogeneous environment.
- Sycara [1998] lists 6 challenges of multiagent systems:
  - How to decompose problems and allocate tasks to individual agents.
  - How to coordinate agent control and communications.
  - How to make multiple agents act in a coherent manner.
  - How to make individual agents reason about other agents and the state of coordination.
  - How to reconcile conflicting goals between coordinating agents.
  - How to engineer practical multiagent systems.

Multiagent Systems Engineering (MaSE) is an attempt to answer how to engineer practical multiagent systems, and to provide a framework for solving multiagent problems [DeLoach 1999].

- Domain level design
- Agent level design
- Component design
- System design

Multiagent Systems Engineering (MaSE) [DeLoach 1999] uses its own agent modeling language.

- Agent Diagram
- Communication Hierarchy Diagram
- Communication Class Diagram
- Deployment Diagram

More Agents

- More definitions and examples found in Franklin and Graesser [1996].
- Application examples found in Maes [2003].
- Theory and application for agents found in Wooldridge, M. and Jennings, [June 1995].
- News and information for agents found in the UMBC Agent Web [UMBC 2003].
- Research examples found in the ABE Group, Stanford Univ., [ABE 2003].


- BDI agent-based software modeling techniques
- BDI agent-based programming languages

References

- Acknowledgement
  - These notes are summarized mainly from the following references.
References


Hayden, Sandra, Carrick, Christina, and Yang, Qiang, A Catalog of Agent Coordination Patterns, ACM Press, 412-413, 1999.


References